CALCULUS AND LINEAR ALGEBRA IN APL

by James W. England
Swarthmore College
Swarthmore, Pa. 19081

This paper is a report on experimental courses in calculus and linear algebra which have been given for the past three years at Swarthmore College. These courses make up the first year of a two-year introductory mathematics program given by the department. The second year consists of a semester of several variable calculus followed by a "Moore Method" course in introductory analysis. The experimental calculus and linear algebra courses are offered as alternatives to conventional courses in the same subjects. In the conventional courses and the experimental courses we stress the constructive and computational features of calculus and linear algebra.

In the experimental courses, which are the subjects of this report we use computing in a central way: to represent the ideas and concepts of calculus and linear algebra and facilitate their analysis and exploration. To do this we use APL (A Programming Language) as the notation. In doing so we use the advantages of the APL notation to alter the presentation of a number of topics contained in calculus and linear algebra. The advantages we see in the APL notation are its relative conciseness, its lack of ambiguity, its ability to handle arrays in a straight forward manner and its demand for explicitly defined functions.

A result of using the APL notation in these courses is a high degree of integration of the subjects of linear algebra and calculus. This notation makes it possible for the students to easily use the computer as an experimental device. This is possible because the

notation which is used to express the concepts and techniques which arise in the courses is capable of being used to direct a machine to implement those ideas.

In general, the notation one uses for a subject tends to grow from the ideas in the subject. That is, topics considered important are easy to express while those considered less important become difficult to state. However, in using APL as the notation for calculus and linear algebra we are faced with the reverse situation. That is, we are using a new notation for well established subjects. It is, therefore, important to remember that the notation (or language) one uses to describe a subject will shape one's view of the subject. There is a danger that important ideas may not be easy to express. Of course, there is also the potential that previously difficult and important ideas are easier to express and hence more accessible. Hence we have tried to identify the effect of using APL as the notation in these courses. APL as the notation for calculus and linear algebra causes one to stress the constructive and computational results of the subjects. Many of the analytic tools, algorithms and results of theorems are described as explicit functions. Once so defined they can easily be experimented with in order to gain a clearer understanding of their meanings and effect. They also remain available for use throughout the remainder of the course. Every function considered in these courses is introduced by at least one explicit construction. In addition to an understanding of the topics covered in the course we think the students leave the course with a rather different view of mathematics. This view includes seeing experimentation as a part of mathematics.

The demand for explicitly defined functions stresses a constructive view of these subjects and probably results in the students taking a similar view of mathematics. The nature of this program is one in which the notion of approximation is fundamental. Generally we proceed from a class of well understood functions to a new class of functions via approximation. In this way students see the idea of approximation used to define new objects as well as approximating known objects for computational purposes.

Polynomials are studied extensively in the calculus course. A polynomial is identified with its coefficient vector. Functions defined on polynomials are therefore defined on vectors. A large number of these functions are linear. This makes it useful to introduce the notion of linear functons and their matrix representation together with some elementary matrix algebra. For example, the function *POLY* is a function of two arguments, the left argument being the coefficient vector of a polynomial, and the right argument being a vector consisting of the numbers at which the polynomial is to be evaluated. For example, (¯4,0,1) *POLY* (¯2,0,2,4) results in (0,¯4,0,12). The polynomial associated with (¯4,0,1) in standard notation is $¯4+X^2$. When the right argument is held fixed, say *V*, the function *POLY V* is linear in its left argument. Therefore, it has a matrix representation, the size of which depends upon the degree of polynomials in the left domain. For polynomials of degree 2 and *V*←(¯2,0,2,4) the matrix representation of *POLY V*

is $B\leftarrow$

$$\begin{matrix} 1 & ¯2 \\ 0 & 0 \\ 1 & 2 \\ 1 & 4 \end{matrix}$$

and for the polynomials of degree 4, it is

$A\leftarrow$

$$\begin{matrix} 1 & ¯2 & 4 & ¯8 \\ 1 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 \\ 1 & 4 & 16 & 64 \end{matrix}$$

We determine these matrices in the standard way. That is, we apply the function *POLY V* to the columns of the *N* by *N* identity matrix, *ID N*. The name of the function which does this is *REP* so that '*POLY* (¯2,0,2,4)' *REP ID* 4 produces the matrix *A* as given above. Notice that for a vector *V* if we let *ρV* denoted the size of *V* (in this case, the number of elements in *V*), then '*POLY V*' *REP ID ρV* is a Vandermonde matrix. Provided the entries in *V* are distinct this matrix is easily seen to be invertible. From this we can conclude that given *N* points in the plane, the first coordinates of which are distinct, there exists a unique polynomial of degree at most *N*-1 which passes through these points. In fact, we can now easily define a function, which we will call *CFP*, with the property that if *P* and *Q* are vectors of the same length and the entries in *P* are distinct then *P CFP Q* is the coefficient vector of the unique polynomial of degree at most ¯1+*ρV* which passes through the points whose coordinates are determined by *P* and *Q* . In order to complete the description of the function *CEP* it is necessary to give a brief introduction to the APL notation.

An expression written in APL is computed by proceeding from right to left. Most parentheses I will use are unnecessary but should help the reader unfamiliar with APL notation. Most functions I define have a much more general definition than what I give! That is, the following at best is an introduction to the parts of the notation necessary to understand the concepts discussed in this paper. The operator ∘. is monadic (has one argument written on the right) with the argument of ∘. being a dyadic function, e.g. +,-,×,÷, etc. If *F* denotes a dyadic function ∘.*F* is a matrix whose [*I*;*J*] element is the

result of applying F to the I^{th} element of the left argument and the J^{th} element of the right argument. For example (0,1,2)∘.+(5,6) results in

```
5 6
6 7
7 8.
```

The symbol * denotes the power function and ⍳ is a monadic function defined on the positive integers with ⍳N resulting in the first N non-negative integers. For example, 8=2*3, 9=3*2 and (0,1,2,3)=⍳4. Recall from the last paragraph ⍴ is a monadic function which results in the size of its argument. Thus, 3=⍴(7,2,6) and (3,2)=⍴((0,1,2)∘.+(5,6).

Returning to the above example let V←(¯2,0,2,4) and notice that V∘.*(⍳(⍴V))

```
1  ¯2   4   8
1   0   0   0
1   2   4   8
1   4  16  64
```

The function denoted by +.× (called the inner product) is dyadic. When both the left and right arguments are matrices of the proper size, the result is the matrix product. When the left argument is a matrix and the right argument a vector the result is the standard inner product of a matrix and vector.

Let C be a vector satisfying (⍴C)=(⍴V). That is, the degree of the polynomial associated with C is at most ¯1+(⍴V). Then (V∘.*(⍳(⍴C)))+.×C is the value of the polynomial C at the elements of V. The last statement can serve as a definition of the function *POLY*. We return (at last) to the definition of the function *CFF*. Given P and Q as above we want a polynomial C, with (⍴C)=(⍴P)

satisfying $Q=(C\ POLY\ P)$. Written in matrix form we want to solve the system of linear equations $Q=(P\circ.*(\iota(\rho C)))+.\times C$. Notice that since I am only interested in polynomials satisfying $(\rho C)=(\rho P)$ I can replace (ρC) by ρP in the definition of *POLY*. Hence, we can rewrite the above equations as $Q=(P\circ.*(\iota(\rho P)))+.\times C$. This system of equations has a unique solution if and only if its matrix of coefficients, $P\circ.*(\iota(\rho P))$ is non-singular. Parenthetically, note that any easy way to see that the Vandermode matrix is non-singular using the definition of *POLY* is that the unique polynomial of degrees less than N which is zero at N distinct numbers is the zero polynomial. Finally, if A is a non-singular matrix, $\boxminus A$ denotes the inverse of A. When $\boxminus$ is used dyadically with a non-singular matrix, A, a right argument and an appropriate vector or matrix, V, as a left argument the result is $(\boxminus A)+.\times V$. $\{(V\boxminus A)=(\boxminus A+.\times V)\}$. Thus we can now define the function *P CFP Q* by $Q\boxminus(P\circ.*(\iota\rho P))$. The function *CFP* is now available for use throughout the remainder of the course.

The following example may help explain some of the points I have raised. The example is an outline of how numerical integration can be presented. To describe this I need to introduce a few additional APL functions.

Scalar functions extend to arrays element by element. For example, $(6,4,3,5)=((6,8,9,20)\div(1+\iota 4))$. The symbol ϕ denotes the function which reverses the order of the elements of a vector, e.g. $(3,2,1,0)=\phi(\iota 4)$. The reduction operator, denoted by /, take a dyadic function as its left argument and a vector as its right argument and returns the value of the expression obtained by inserting the left argument between succeeding element of the right argument. For

example 6=+/(0,1,2,3) and (.5)=÷/(1,2). If *V* is a vector and *I* is an array whose elements are a subset of the indices of *V* then *V*[*I*] is the array formed by replacing each element of *I* by the element in *V* whose index it is. The indices of a vector *V* are ⍳(⍴*V*). Recall that ⍳*N* includes zero. Let *V*←(2,3,6,8), *V*[0]=2; *V*[2]=6; *V*[3]=8; *V*[2 0 3]=(6,2,8) and if

$$I\leftarrow \begin{matrix} 1 & 3 \\ 2 & 0 \\ 3 & 2 \end{matrix} \quad \text{then } V[I] \text{ is } \begin{matrix} 3 & 8 \\ 6 & 2 \\ 8 & 6 \end{matrix}$$

We can now give a description of numerical integration. At the end of this paper we give numerical examples of the functions we will define. The reader may wish to consult these examples while reading this section. The function *PII* (polynomial indefinite integral) is defined on the coefficient vector of a polynomial *C*, by the expression (*0*,*C*÷(1+⍳(⍴*C*))) so that *PII C* is an indefinite integral of the polynomial *C*. If *C* is a polynomial and if *V* is a vector whose entries are the endpoints of an interval then the expression -/((*PII C*) *POLY* (⌽*V*)) will define a dyadic function written *V PDI C* (polynomial definite integral) which produces the definite integral of *C* over the interval *V*. Assume the function *N* is integrable on the interval *V*←(*A*,*B*). For *N* a positive integer *P*←*A*+(((*B*-*A*)÷*N*)×⍳(*N*+1)) is a partition of *V* into *N* subintervals of equal length. With these functions we can define a (bad) approximation to the integral of *F* over the interval *V* by the expression

(*P*[0,¯1+(⍴*P*)]) *PDI* (*P CFP* (*FP*)) (*)

The difficulty with this approximation is that in order for it to be reasonably accurate over a large interval we must take *N* to be quite large. In turn, this means the matrix which must be inverted in

CFP is quite large. Hence, we need a more sophisticated technique.

Recall that a K^{th} order approximate integral of F over V (K=2 for the Trapezoid Rule, K=3 for Simpson's Rule) is obtained by adding the integrals of the $(N-1)\div(K-1)$ polynomials obtained by fitting polynomials to K successive points of P and FP. Now, if in expression (*) we replace FP by a vector Q with $(\rho P)=(\rho Q)$ the resulting expression defines a function of P and Q. With P held fixed, this is linear in Q. We denote this function by *AI*. That is, *P AI Q* is defined by *(P[0,¯1+(ρP)]) PDI (P CFP Q)*. The function *COEFF* is defined so that it sends the partition P to the matrix representation of the linear function *P AI*. The expression *'P AI' REP ID (ρP)* will serve as a suitable definition for *COEFF P*. We have *((COEFF P)+.×Q)=P AI Q*. We need the following identity, which is straightforward to prove. For A and S scalars and P a partition *(COEFF S×(A+P))=(S×COEFF P)*. Thus for an equally spaced partition P we have *(COEFF P)=(S×COEFF(⍳ρP))* where S is the distance between adjacent point of P. Finally, to easily lump together successive elements of a partition into groups of K we need only produce the proper matrix of indices. As long as $(N-1)\div(K-1)$ is a positive integer this is given by $(\iota K)\circ.+(K-1)\times\iota((N-1)\div(K-1))$. The dyadic function *LUMP* is defined by letting *K LUMP P* equal *P[(⍳K)∘.+(((K-1)×⍳(ρP)-1)÷(K-1))]*.

To finish this problem, recall F is defined on the interval V, P is a partition of this interval and K is chosen so that $((\rho P)-1)\div(K-1)$ is an integer. If *M←K LUMP FP* then the 0^{th} column of *M*, *M[;0]* is the value of F at the first K point of P so that *(COEFF P[⍳K])+.×M[;0]* is *P[⍳K] AI M[;0]* or the value of the integral of the polynomial fit to the first K points with X-coordinates P and Y-coordinates FP.

We can simplify this by the identity we gave above to $S\times(COEFF\ \iota K)+.\times M[;0]$ where S is the distance between adjacent points of P. This same argument holds for each group of K points, which corresponds to the various columns of M. Thus $S\times(+/(COEFF\ \iota K)+.\times(K\ LUMP\ (FP))$ is the value of a K^{th} order approximate integral of the function F over the interval with respect to the partition P. We conclude by having the students define a dyadic function *INT* so that (A,B) *INT* 'F' gives an approximate value for the integral of the function F over the interval $(A\leq X)\wedge(X\leq B)$. The function *INT* (integral) is defined early in the calculus course. It is used to compute definite integrals until techniques of integration are covered.

The sections dealing with series is primarily devoted to power series. Series of numbers are considered only to check the convergence of power series at the endpoints of the interval of convergence. Topics in differentiation are covered in a rather standard way. However, since APL allows one to easily manipulate arrays the students define a symbolic differentiator; that is, a function whose argument is a symbol string which represents an elementary function and whose result is a symbol string representing the derivative of the original function.

The text we use is Calculus in a New Key by D. L. Orth, A.P.L. Press, Swarthmore, Pa. As far as we know, this is currently the only text available. The discussion we gave of numerical integration is, in a different form, contained in Orth's book.

Courses such as these, involving a rather radical departure from the standard courses in calculus and linear algebra can create

certain difficulties. The first and ultimately the most serious difficulty is its acceptance within the local scientific community. Calculus and linear algebra are subjects which are fundamental to the physical and social sciences. Hence any departure from the beaten path is suspect. At the same time much elementary mathematics is learned by students as they attempt to use it in other science courses. Thus, there is a danger that students will not get reinforcement of their calculus and linear algebra in other courses. This problem seems to be one which can be decreased in its severity by a careful job of explaining the program to other departments. The precision of APL sometimes makes it difficult for people to use it informally, particularly in lectures. Our attitude in this regard has been to use the notation informally once it is understood by the students. Related to this is the fact that people often have difficulty looking at an APL expression and have it be as meaningful as a similar expression in conventional notation. This problem lessens as the course proceeds and people become more familiar with the APL notation.

Our experience with this program is such that we are attempting, whenever possible, to use the APL notation in other mathematics courses. In one sense this is an attempt to see to what extent APL can be used as a notation for mathematics. We hope that others will join us in this work.

Examples of the functions used in numerical integration

(1) *PII C* (Polynomial indefinite integral)

```
(0,(C÷1+⍳(⍴C)))
```

Example: *PII* (3,8,12)

```
(0,3,4,4)
```

(2) (*A*,*B*) *PDI C* (Polynomial definite integral)

```
-/(PII C) POLY (B,A)
((PII C) POLY B)-((PII C) POLY A)
```

Example: (0,3) *PDI* (3,8,12)

```
((0,3,4,4) POLY 3)-((0,3,4,4) POLY 0)
153 - 0
153
```

(3)

```
P←(¯3,¯2,6,¯2.2,¯1.8,-1.4,¯1,¯.6,¯.2,.2,
   .6,1,1.4,1.8,2.2,2.6,3)
```

P is a partition of the interval $(\bar{}3 \leq X) \wedge X \leq 3$.

```
P[0,¯1+⍴P]=(¯3,3)
```

F is the function defined by

$(*-(X*2)\div 2)\div(o2)*.5 \leftrightarrow \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{X^2}{2}\right)$

```
F = (.004,.014,.035,.079,.150,.242,.333,.391,
     .391,.333,.242.,150,.079,.035,.014.,004)
P AI Q
(P[0,¯1+(⍴P)]) PDI (P CFP Q)
P AI FP
.99744
```

(4) A table of values of *COEFF* ⍳*N* for various values of *N* is given below:

N	*COEFF* ⍳*N*
2	(1,1)÷2
3	(1,4,1)÷3
4	(3,9,9,3)÷8
6	(475,1875,1250,1250,1875,475)÷1440

(5) Letting *K*=6 we get

```
      2↑6 LUMP P                    2↑6 LUMP F P
¯3.00 ¯1.00  1.00            .00  .24  .24
¯2.60  ¯.60  1.40            .01  .33  .15
¯2.20  ¯.20  1.80            .04  .39  .08
¯1.80   .20  2.20            .08  .39  .04
¯1.40   .60  2.60            .15  .33  .01
¯1.00  1.00  3.00            .24  .24  .00
```

```
S←P[1]-P[0]
```

The approximate value of the intergral of *F* over (¯3≤*X*)∧*X*≤3 using 5th degree polynomials is

```
S×+/(COEFF ⍳6)+.×6 LUMP F P
```

```
0.9971193436
```

Letting *K*=4 we get

```
      2↑4 LUMP P                      2↑4 LUMP F P
¯3.00 ¯1.80 ¯.60  .60 1.80      .00 .08 .33 .33 .08
¯2.80 ¯1.40 ¯.20 1.00 2.20      .01 .15 .39 .24 .04
¯2.20 ¯1.00  .20 1.40 2.60      .04 .24 .39 .15 .01
¯1.80  ¯.60  .60 1.80 3.00      .08 .33 .33 .03 .00
```

The approximate value of the integral of *F* over the same interval using 3rd degree polynomials is

```
S×+/(COEFF ⍳4)+.×4 LUMP F P
```

```
0.9972530128
```